LA-UR-99-2646

Title: **Model Inversion Using Bayesian Inference And Genetic Algorithms Part II: A Simplified Born-Mayer Potential**

Author(s): Brian J. Reardon, MST-6

Submitted to:

http://lib-www.lanl.gov/la-pubs/00326796.pdf

Los Alamos
NATIONAL LABORATORY

Form No. 836 R5
ST 2629 10/91

Model Inversion Using Bayesian Inference And Genetic Algorithms Part II: A Simplified Born-Mayer Potential

Brian J. Reardon, MST-6, Los Alamos National Laboratory, Los Alamos, NM 87545

Abstract

This work explores the use of a genetic algorithm (GA) in conjunction with Bayesian inference to optimize the parameters of a simplified Born-Mayer potential function. The efficiency of the GA and the accuracy of the information provided by the Bayesian analysis were examined by varying the number of objectives and parameters. The number of objectives was varied by taking higher order derivatives of the potential function. Likewise, the number of variables was changed by adding a dummy variable to the gene sequence. This variable is not implemented in the objectives themselves. Rather, it serves as a measure of performance for the GA and the Bayesian analysis.

The results of this work indicate that both the GA and the Bayesian analysis techniques can easily handle large numbers of objectives. The implementation of a dummy variable has an effect on the selection procedure during the niching operation. This results in the dummy variable appearing to be more sensitive or influential than other parameters actually used in the objective functions. The apparent sensitivities of the parameters as obtained from principle component analysis (PCA) were confirmed by the evolution of the expectation values and standard deviations.

1.0 Introduction

1.1 Inverse and Ill Posed Problems in Materials Science and Engineering

There is an ever increasing need in materials science and engineering to fit the parameters of models, which are to be used in a predictive capacity, using underdetermined experimental data sets. Model inversion of this type falls under the general category of inverse and ill – posed problems and can often be cast into the framework of Bayesian statistics (Tarantola, 1987). Such problems include determining powder densification models from limited density data, chemical potential determination from limited phase diagram data containing a high degree of uncertainty, and mechanical threshold strength determination from mechanical tests also with a high degree of uncertainty. In all of these examples, model parameters must be optimized using limited and uncertain data sets that leave the inversion underdetermined. Likewise, if the models are to be used in a predictive capacity, there is a need to be able to quantify the expected deviation of the model from reality.

This report shows how a fuzzy logic based multi-objective genetic algorithm (GA) (Reardon 1999) can be used as a Bayesian Inference Engine (BIE) to evolve a posterior probability density (PPD) of the model parameter vector space:

$$M_i = \{m_1, m_2, m_3, \ldots, m_N\}^T \qquad \text{Eq. 1}$$

where M_I is a particular model to be tested, m_J is one of the N parameters used in the model and T signifies the transpose of the vector. The GA evolves a set or population of M_I's which effectively defines the PPD. Once the PPD has been sufficiently determined by the GA, parameter vectors are selected and used in the physics of the forward problem, for future experimental conditions, to evaluate the predictive capacity of the model.

The ability of the GA and Bayesian statistics to handle multiple variables and objectives will be addressed here with a simplified Born-Mayer interatomic potential. The potential function has the form:

$$\Phi = \frac{1}{r} + A * \exp\left(\frac{\sigma - r}{d}\right) \qquad \text{Eq. 2}$$

where r=1.17288 is the interatomic distance, A is the pre-exponential factor, σ is the sum of the two radii of the atomic pair, and d is the hardness parameter. When A=4.0, σ=0.4, and d=4.0 the derivatives of Eq. 2 have the values:

$$\begin{aligned}
&\Phi = -0.67087 \\
&\frac{\partial \Phi}{\partial r} = 4.9085E-6 \\
&\frac{\partial^2 \Phi}{\partial r^2} = 1.6681 \\
&\frac{\partial^3 \Phi}{\partial r^3} = -8.4602 \\
&\frac{\partial^4 \Phi}{\partial r^4} = 35.710 \qquad \text{Eq. 3} \\
&\frac{\partial^5 \Phi}{\partial r^5} = -139.99 \\
&\frac{\partial^6 \Phi}{\partial r^6} = 508.56 \\
&\frac{\partial^7 \Phi}{\partial r^7} = -1570.1 \\
&\frac{\partial^8 \Phi}{\partial r^8} = 2310.7
\end{aligned}$$

The goal of this study is to determine the efficiency to which the GA finds the values of A, σ, and d given that the objective values of Eq. 3 are known to ±1%. The number of objectives used in the optimization will be systematically increased to evaluate the GA's ability to handle large numbers of objectives. Furthermore, a dummy variable will be

incorporated into the optimization to determine the effects of increasing the number of variables and to evaluate the GA's ability to overcome genetic drift. Thus, the model parameter vector will have the form:

$$M_i = \{A_i, \sigma_i, d_I, R_i\}^T \qquad \text{Eq. 4}$$

where $3.5 \le A \le 8.0$, $0.395 \le \sigma \le 0.41$, $3.5 \le d \le 8.0$, and $0.0 \le R \le 1.0$. The choice of ranges for A, σ, and d where selected based on the fact that σ is often well known from diffraction data and other experiments whereas A and d are not experimentally accessible.

1.2 Bayesian Statistics in Model Inversion

Consider a model parameter vector such as the one defined in Eq. 4 and also consider a data vector defined as:

$$D = \left\{ d_1, d_2, d_3, \ldots, d_{N_D} \right\}^T \qquad \text{Eq. 5}$$

where N_D is the total number of experimentally derived data points. The goal of Bayesian analysis is to come up with a way of accepting or rejecting a particular model (M) or hypothesis given an experimental data set (D) and prior knowledge about the problem. Thus, in Bayesian statistics, the model or hypothesis is assigned a probability of acceptance and the total probability distribution function (PDF) of a series of models being tested makes up what is commonly called the posterior probability density (PPD). This goal is achievable through the central tenant of Bayesian statistics, Bayes' Theorem:

$$\sigma(M \mid D) = \frac{P(M,D)}{P(D)} = \frac{P(D \mid M)P(M)}{\int P(D,M)dM} \qquad \text{Eq. 6}$$

which is essentially the definition of conditional probability. This rule was first proposed by Rev. Thomas Bayes and published posthumously in 1763 but has been ignored up

until the last 20 years due to the computational difficulties in evaluating the probability integrals (Bayes, 1763). This theorem says that the conditional probability of a model being correct given a set of data is a ratio of the PDF of M and D to the PDF of D alone. The term P(D | M) is not a PDF but a likelihood function. Thus, while the individual components of P(D | M) are probabilities, the function itself does not integrate to 1.0.

Bayes' rule as written above differs considerably from classical frequentist statistics because of the dependence of the PPD on the prior PDF, P(M). P(M) often contains subjective information about the problem that the experimentalist has *a priori*. Another major departure from frequentist statistics is the way the PPD is updated as new experimental data becomes available. The frequentist view point is that P(D) should be considered an unchanging distribution and also that it is inappropriate to try to assign a probability of correctness to a hypothesis.

Consequently, Bayes' Rule provides the scientist with a tool that classical statistics is not capable or providing, namely, a mathematical formalization of the scientific method. When a phenomenon is observed, a hypothesis explaining the event is created often with the observer's own bias and experience in mind. This hypothesis is then tested against new experimental data and if the data supports the hypothesis then the belief in or probability of acceptance of the hypothesis increases. An excellent introduction to the Bayesian approach to hypothesis testing can be found in Chapter 4 of Antelman (1997).

The main difficulty in using Bayes' rule, lays in the evaluation of the denominator:

$$P(D) = \int P(D,M)dM, \qquad \text{Eq. 7}$$

where the integral is formally carried over the entire N-dimensional model parameter space. The accurate and fast approximation of the integration of these N-dimensional, discontinuous PDF's is the topic of many papers. Duijndam (1988a, 1988b) discussed the used of Bayes' Rule in model inversion and accomplished the above integration by assuming the PPD had a Gaussian shape then optimized the Gaussian parameters

using least squares. Unfortunately, most PPD's are not Gaussian in nature and thus other techniques were needed. These techniques include Monte Carlo integration, Gibb's Sampling, and genetic algorithms (Sen and Stoffa ,1992, 1996; Sen et al., 1993; Mallick, 1995; Gerstoft, 1998).

The PPD is a difficult function to visualize due to its multidimensionality and its change with every new experimental data point. However, once the PPD is derived, regardless of the method, a number of important parameters describing it can be easily calculated.

The mean model can be calculated using:

$$\langle M \rangle = \int M \sigma(M \mid D) dM \qquad \text{Eq. 8}$$

Likewise, the *a posteriori* model covariance matrix is given by:

$$C_M = \int (M - \langle M \rangle)(M - \langle M \rangle)^T \sigma(M \mid D) dM. \qquad \text{Eq. 9}$$

The covariance matrix provides a number of useful parameters. The standard deviation associated with the mean model is obtained through the square roots of the diagonal elements of C_M. Normalization of C_M through:

$$C_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}}\sqrt{C_{jj}}} \qquad \text{Eq. 10}$$

produces the correlation matrix.

With C_M determined, a principle component analysis (PCA) will provide valuable insight on how well the GA is converging and what model parameters are most significant or sensitive. In PCA the C_M is transformed into a new set of axes of the same number which are orthogonal to each other and are ordered based on the variance associated with that axis. The principle components of C_M can be obtain by

computing its set of eigenvalues (Λ) and corresponding orthogonal eigenvectors (U) such that:

$$C_M = U\Lambda U^T \qquad \text{Eq. 11}$$

is satisfied. In a d-dimensional variable space there are d eigenvalues or principle components. However, many principle components may have small variances and thus the intrinsic dimensionality of C_M is k where k<d.

In the context of a PPD evolved by a GA, PCA is a powerful tool that assists in overcoming many deficiencies in GA's. First, as the population evolves, the sum of the eigenvalues of C_M approach a limit. When the rate of convergence reaches an acceptable minimum the GA can be stopped. Second, the largest eigenvalues and their corresponding eigenvectors indicate the most significant variables or groups of variables in the model given the available data. Thus, PCA provides a sensitivity analysis for the variables in the model.

Once a PPD has been determined to be reliable based on the stabilization of the eigenvalues, an optimum model can be selected.

1.3 Genetic Algorithms in Model Inversion and Parameter Optimization

A detailed account of how a GA operates has been provided elsewhere (Reardon 1998a, 1998b, 1999). In short, a GA randomly generates a set or population of parameter vectors M_i's where i = 1 to N and N is the population size. This initial selection, which occurs within parameter ranges set by the user, constitutes the *a priori* information used in Bayes' Theorem. From this set, parameter vectors that satisfactorily solve the optimization problem are selected. The selected members, which are each defined by a haploid binary string, exchange string components and thus create new members. The bits of the new member's strings are then randomly flipped with a small degree of probability from 1 to 0 or vice versa. The final members are then inserted into the next generation. Once the next generation is filled the GA starts over with selection, crossover and mutation.

Since the GA acts as a BIE in that it uses Bayes' Theorem to select members in the population for crossover, the output of the GA is the PPD. The generation of a PPD now allows for many of the statistical tools available in Bayesian statistics to be used in the analysis of the output of the GA. Namely from the PPD we can derive <M> and C_M. The beauty of this approach is that the PPD can be generated at virtually no extra cost. Following the method outlined by Sen and Stoffa (1992), a 2-D array of M X B is reserved where M is the number of parameters and B is the number of values each variable can take (i.e. the number of bins). For each model at each generation an unnormalized PPD, σ(M), is computed and stored in the proper position in the bin array for each model parameter comprising each model. At the end of the GA run the model parameter PPD values are normalized. Also in a vector of length M, each component of M σ(M) is stored and summed with the correspond values from the other models. This vector provides <M>. C_M is determined by summing up MM^T σ(M) in a square array of MM for each model and at the end of the run subtracting $<M><M>^T$. The FORTRAN 90 code used to evaluate these quantities was presented previously (Reardon, 1999).

Once the PPD, <M>, and C_M have been sufficiently determined, the GA can be stopped and optimal model parameter vectors can be selected and used in the physics of the forward problem for conditions that have not been experimentally tested.

2.0 Simplified Born-Mayer Potential

Figures 1-4 show the eigenvalues as a function of generation for the optimization problem involving A, σ, d, and R. The corresponding eigenvectors are displayed in Table I. A total of nine optimizations were carried out, each with one additional objective. Thus, the first optimization used only the first objective, the second optimization used the first two objectives and so on. The smallest eigenvalues are shown in figure 1 and correspond to the first eigenvector column of Table I. This eigenvector is dominated by σ for all nine optimizations. Due to the small size of the first eigenvalue compared to the others, it can be concluded that σ is not a dominate or sensitive parameter in this model. This lack of sensitivity is due primarily to the small size of the search range. A larger search range would give a different value for the

sensitivity but in extending the range one is also requiring the GA to traverse a significantly more difficult function landscape. This fact is the reason why previous work (Reardon 1998a) did not show definitive convergence of the parameters. Additionally, since, in the Born-Mayer potential, σ is usually defined as the sum of the radii of two interacting atoms and since this quantity is usually well known, there is no reason to allow it a large search range.

The next smallest eigenvalue distribution is displayed in figure 2. Figure 2 indicates that eigenvalue #2 increases as the number of objectives used in the optimization increases. This would indicate that the parameters which dominate the corresponding eigenvector of eigenvalue #2 also become more sensitive as the number of objectives increases. Inspection of the second column of Table I shows that eigenvalue #2 is dominated by the hardness parameter, d. The d parameter becomes more significant with the number of objectives since each time a derivative is taken the exponential term is multiplied by –d.

The next smallest eigenvalue distribution is displayed in figure 3. Figure 3 indicates that eigenvalue #3 increases slightly as the number of objectives used in the optimization increases and is approximately four times larger than eigenvalue #2. Again, this would indicate that the parameters which dominate the corresponding eigenvector of eigenvalue #3 also become slightly more sensitive as the number of objectives increases. Inspection of the third column of Table I shows that eigenvalue #3 is dominated by the dummy variable, R. This result is rather counter intuitive given that R is not actually used in any of the objective equations. The explanation lies not in the objectives but in the selection routine itself.

In the selection routine, each parameter vector is used in the evaluation of each objective function $f_j(M_i)$ where j = 1 to N_D (N_D: number of objectives or experimental data points). The outcome of the objective function call is then compared to the experimental data using the fuzzy rule set of Figure 5 to obtain a scaled fuzzy fitness value $f_j'(f_j(M_i))$. In Figure 5, D_j is the experimentally observed data point and E_j is the uncertainty associated with D_j. $f_{j:max}$ is the maximum value for objective j in the entire population and

$f_{j:min}$ is the minimum. The fuzzy fitness $f_j'(f_j(M_i))$ is obtained by finding where $f_j(M_i)$ lays on the x axis of Figure 5 and assigning its corresponding y-axis value. The total fuzzy fitness of M_l then is defined as:

$$F_T(M_i) = \frac{1}{N_D}\sum_{j=1}^{N_D} f_j'\left(f_j(M_i)\right) \qquad \text{Eq. 12}$$

where D is the number of objectives. This relation provides the fitness of a model vector as a number between 0 and 1, where 1 is the best fit. The GA selects two members based on their fitness likelihood as defined in figure 5 and Eq. 12. The two selected members are then compared. If one is better fit than the other then the better fit is selected. If they have the same fitness then they are selected according to whoever has the smallest phenotypic crowding factor when compared to the partially filled next generation. In other words, the most unique of the two individuals, when compared to the partially filled next generation, is selected. Thus, as the significant parameters begin to converge, the crowding factor and, by default, selection becomes dominated by R.

The largest eigenvalue distribution is displayed in figure 4. Figure 4 indicates that eigenvalue #4 decreases as the number of objectives used in the optimization increases. This would indicate that the parameters which dominate the corresponding eigenvector of eigenvalue #4 become less sensitive as the number of objectives increases. Inspection of the fourth column of Table I shows that eigenvalue #4 is dominated by the pre-exponential factor, A, and to a smaller extent the hardness parameter, d. The A parameter becomes less significant with the number of objectives since each time a derivative is taken the exponential term is multiplied by –d and thus A loses importance in the overall scheme of things.

Figures 6-9 show the average values of the four variables in the optimization as a function of generation and number of objectives. When three or less objectives are used in the optimization, <A> and <d> tend to converge towards incorrect values.

However, as the number of objectives increases beyond four, the problem becomes less underdetermined and thus the averages for the population converge towards the correct values. The values for <σ> and <R> do not show any clear trends as the number of objectives increases.

Figure 10-13 shows the standard deviations of the four variables in the optimization as a function of generation and number of objectives. For the A parameter, as the number of objectives increases the standard deviation decreases indicating that the variable is converging more easily with more objectives. The d parameter shows, to a smaller extent, the opposite trend. The standard deviations of σ and R do not indicate any clear convergence pattern.

The dummy variable, R, was originally placed in the optimization in order to evaluate the performance of the GA. If the GA is functioning correctly, then the distribution of R should remain random as a function of generation. If, however, the GA is succumbing to genetic drift, then the distribution of R will converge to a single value after a sufficient number of generations. Unfortunately, the randomness of R, in light of the convergence of the other parameters, may have had a detrimental effect on the selection procedure as well as the sensitivity analysis.

To confirm that R did in fact impact the optimization as specified, it was removed and the optimizations rerun. Figures 14-16 show the three eigenvalues for each of the nine optimization conditions with their corresponding eigenvectors displayed in Table II. The smallest eigenvalue (figure 14) shows that for all nine optimization conditions, σ is the most insensitive parameter. The two largest eigenvalues are approximately the same order of magnitude and thus of equal sensitivity. The eigenvectors show that eigenvalue #2 is dominated by the d parameter and to a lesser extent, A, and increases in value as the number of objectives increases. Likewise, eigenvalue #3 is dominated by A and to a lesser extent, d, and decreases as the number of objectives increases.

Figures 17-19 show the average values of the three variables in the optimization as a function of generation and number of objectives. When three or less objectives are used in the optimization, <A> and <d> tend to converge towards incorrect values but

converge to a value closer to 4.0 when 4 or more objectives are used. The value for <σ> does not show any clear trends as the number of objectives increases.

Figure 20-22 shows the standard deviations of the three variables in the optimization as a function of generation and number of objectives. For the A parameter, as the number of objectives increases the standard deviation decreases indicating that the variable is converging more easily with more objectives. The d parameter shows, to a smaller extent, the opposite trend. The standard deviations of σ do not indicate any clear convergence pattern.

The behavior of the expectation values and the standard deviations of the variables confirms the information obtained from the principle component analysis.

3.0 Conclusions

This work explores the use of a genetic algorithm in conjunction with Bayesian inference to optimize the parameters of a simplified Born-Mayer potential function. The efficiency of the GA and the accuracy of the information provided by the Bayesian analysis were examined by varying the number of objectives and parameters. The number of objectives was varied by taking higher order derivatives of the potential function. Likewise, the number of variables was changed by adding a dummy variable to the gene sequence. These variables are not implemented in the objectives themselves. Rather, they serve as a measure of performance for the GA and the Bayesian analysis.

The results of this work indicate that both the GA and the Bayesian analysis techniques can easily handle large numbers of objectives. The implementation of a dummy variable has an effect on the selection procedure during the niching operation. This results in the dummy variable appearing to be more sensitive or influential than either d or σ. The apparent sensitivities of A, σ, and d were confirmed by the evolution of the expectation value and standard deviations of the parameters.

4.0 Acknowledgments

Funded by the Department of Defense, the Department of Energy and Los Alamos National Laboratory which is operated by the University of California under contract number W-7405-ENG-36.

5.0 References

G. Antelman, 1997, “Elementary Bayesian Statistics,” Eds. A. Madansky, R. McCulloch, Edward Elgar Publishing, Inc., Lyme, NH.

T. Bayes, 1763, “An Essay towards solving a problem in the doctrine of Chances,” Philosophical Transactions of the Royal Society, 53, 370-418.

A. J. W. Duijndam, 1988a, “Bayesian Estimation in Seismic Inversion. Part I: Principles,” Geophysical Prospecting, 36, 878-898.

A. J. W. Duijndam, 1988b, “Bayesian Estimation in Seismic Inversion. Part II: Uncertainty Analysis,” Geophysical Prospecting, 36, 899-918.

P. Gerstoft and C. F. Mecklenbräuker, 1998, “Ocean Acoustic Inversion with Estimation of *a posteriori* probability distributions,” Journal of the Acoustical Society of America, 104, 2, 808-819.

S. Mallick, 1995, “Model-based Inversion of Amplitude-variations-with-offset Data Using a Genetic Algorithm,” Geophysics, 60, 4, 939-954.

B. J. Reardon, 1998a, “Fuzzy Logic Vs. Niched Pareto Multiobjective Genetic Algorithm Optimization,” Modeling and Simulation in Materials Science and Engineering, 6, 717-734, 1998, http://www.iop.org/Journals/ms

B. J. Reardon, 1998b, “Optimization of Densification Modeling Parameters of Beryllium Powder Using a Fuzzy Logic Based Multiobjective Genetic Algorithm,” Modeling and Simulation in Materials Science and Engineering, 6, 735-746, http://www.iop.org/Journals/ms

B. J. Reardon, 1999, “Model Inversion Using Bayesian Inference and Genetic Algorithms,” LA-UR-99-1980, Los Alamos National Laboratory, Los Alamos, NM 87545, April, http://lib-www.lanl.gov//la-pubs/00326698.pdf.

M. K. Sen, B. B. Bhattacharya, P. L. Stoffa, 1993, “Nonlinear inversion of resistively sounding data,” Geophysics, 58, 4, 496-507.

M. K. Sen and P. L. Stoffa, 1992, “Rapid sampling of model space using genetic algorithms: Examples from seismic waveform inversion,” Geophysics Journal International, 108, 281-292.

M. K. Sen and P. L. Stoffa, 1996, “Bayesian inference, Gibb’s sampler and uncertainty estimation in geophysical inversion,” Geophysical Prospecting, 44, 313-350.

A. Tarantola, 1987, “Inverse Problem Theory, Methods for Data Fitting and Parameter Estimation,” Elsevier, Amsterdam.

6.0 Tables

Table I. The final eigenvectors for the four variable optimization trial with increasing numbers of objectives.

# Objs.	Var. ID	λ_1	λ_2	λ_3	λ_4
1	A	0.0018080	0.15165	-0.0063404	0.98841
	σ	0.99999	-0.0049685	-7.1028e-05	-0.0010674
	d	-0.0047492	-0.98842	-0.0033035	0.15163
	R	6.6802e-05	-0.0023042	0.99998	0.0067679
2	A	0.00094479	-0.18902	0.0076483	0.98194
	σ	-1.00000	0.0017681	-0.0010057	0.0013103
	d	0.0019796	0.98197	0.0040810	0.18899
	R	0.0010210	0.0025601	-0.99996	0.0082804
3	A	-0.0012802	0.33336	0.044649	0.94174
	σ	1.00000	-0.0016035	-0.00031737	0.0019421
	d	-0.0021588	-0.94271	0.0024278	0.33359
	R	-0.00038015	0.012609	-0.99900	0.042900
4	A	0.0018010	0.42528	-0.015589	0.90492
	σ	1.00000	-0.00067347	0.00028663	-0.0016688
	d	0.00010083	-0.90505	-0.0023383	0.42530
	R	0.00025835	-0.0045142	-0.99988	-0.015104
5	A	-0.0015163	0.57530	0.0083127	0.81790
	σ	1.00000	0.00065699	0.00018116	0.0013900
	d	-0.00026663	-0.81778	0.025851	0.57495
	R	0.00016172	-0.016364	-0.99963	0.021670
6	A	0.0041770	0.56878	-0.020787	0.82222
	σ	0.99999	-0.0039264	-0.00032007	-0.0023720
	d	-0.0018921	-0.82234	-0.032783	0.56805
	R	-0.00034513	0.015148	-0.99925	-0.035740
7	A	0.00090125	0.54444	-0.012930	0.83870
	σ	1.0000	-0.00054004	-0.00021272	-0.00072730
	d	-5.9178e-05	-0.83880	-0.0096493	0.54436
	R	-0.00022383	0.0010545	-0.99987	-0.016099
8	A	-0.0013767	-0.43313	-0.047273	0.90009
	σ	-1.00000	0.0012071	0.00030535	-0.00093267
	d	0.00067360	0.90122	-0.038141	0.43167
	R	0.00026645	0.013924	0.99815	0.059124
9	A	0.00043635	0.43938	-0.024475	0.89797
	σ	0.99999	-0.0036610	-0.00048889	0.0012920
	d	-0.0038664	-0.89825	-0.022155	0.43891
	R	-0.00041413	0.0091540	-0.99945	-0.031720

Table II. The final eigenvectors for the three variable optimization trial with increasing numbers of objectives.

# Objs.	Var. ID	λ_1	λ_2	λ_3
1	A	-0.0010144	-0.12736	0.99186
	σ	-0.99999	0.0029582	-0.00064298
	d	0.0028522	0.99185	0.12736
2	A	0.00090144	-0.18009	0.98365
	σ	-0.99999	0.0039655	0.0016426
	d	0.0041965	0.98364	0.18009
3	A	0.00077103	-0.28170	0.95950
	σ	-0.99999	0.0034499	0.0018164
	d	0.0038218	0.95950	0.28169
4	A	0.0015669	0.41126	0.91152
	σ	1.00000	-0.00067776	-0.0014130
	d	-3.6698e-05	-0.91152	0.41126
5	A	0.00042968	0.55127	0.83433
	σ	1.0000	-0.00018421	-0.00039322
	d	6.3072e-05	-0.83433	0.55127
6	A	-0.0012382	-0.54771	0.83667
	σ	-1.00000	0.0014703	-0.00051750
	d	0.00094674	0.83667	0.54771
7	A	-0.0023353	-0.52101	0.85355
	σ	-1.00000	0.0023833	-0.0012809
	d	0.0013669	0.85355	0.52101
8	A	-0.0013972	-0.46967	0.88284
	σ	-1.00000	0.0021772	-0.00042427
	d	0.0017229	0.88284	0.46967
9	A	-0.0023313	-0.42281	0.90622
	σ	-0.99999	0.0030611	-0.0011444
	d	0.0022902	0.90622	0.42281

7.0 Figures

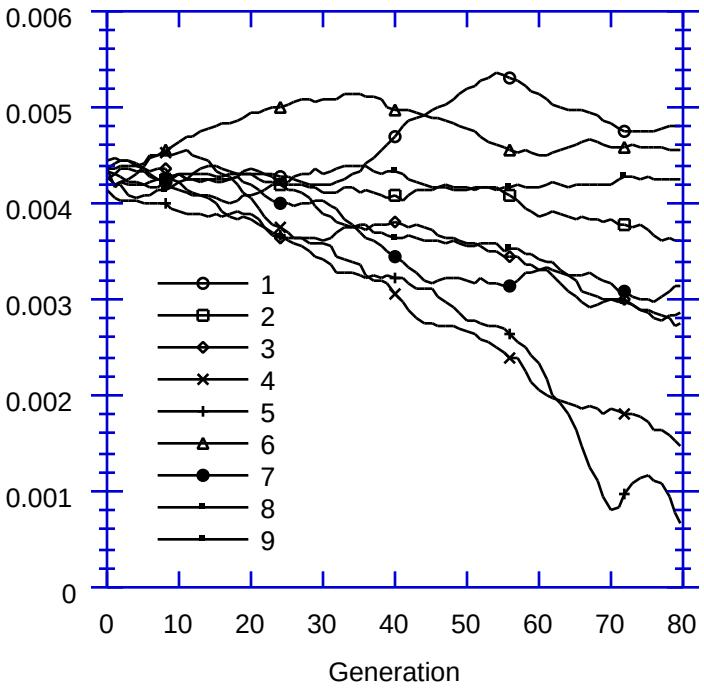

Figure 1. Eigenvalue #1, λ_1, as a function of generation and number of objectives.

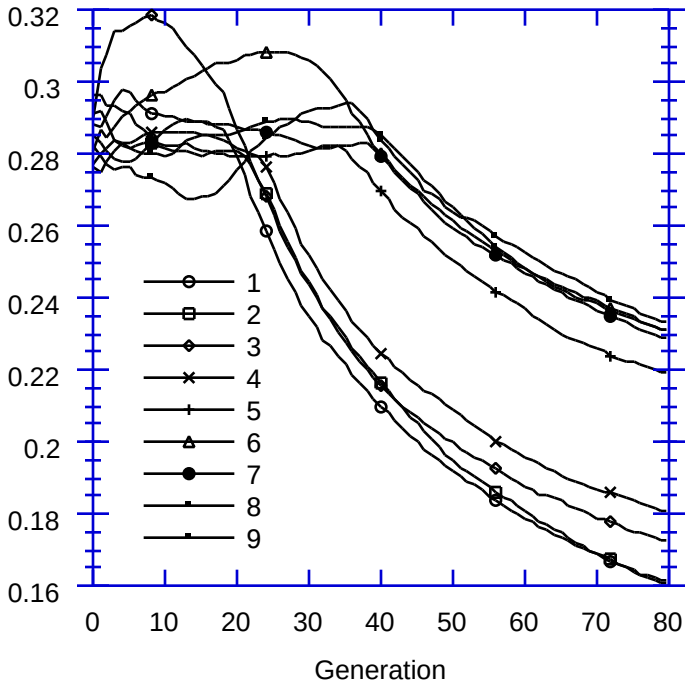

Figure 2. Eigenvalue #2, λ_2, as a function of generation and number of objectives.

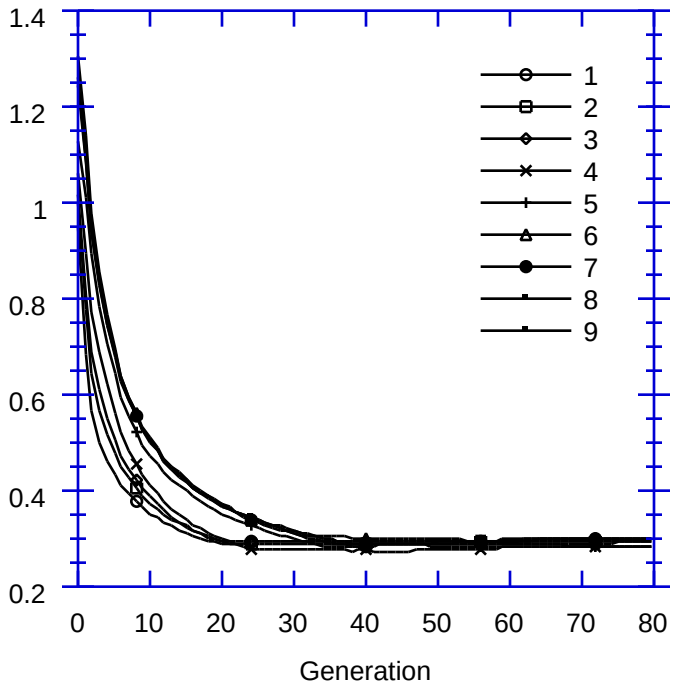

Figure 3. Eigenvalue #3, λ_3, as a function of generation and number of objectives.

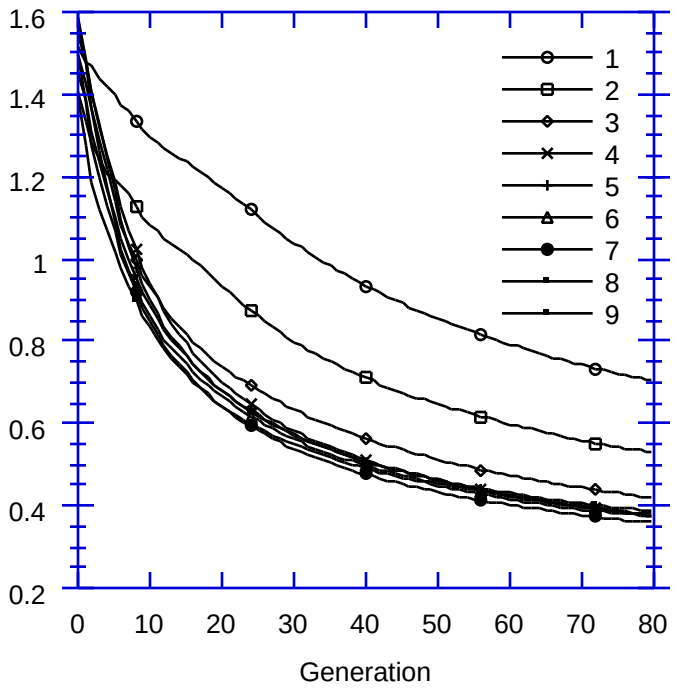

Figure 4. Eigenvalue #4, λ_4, as a function of generation and number of objectives.

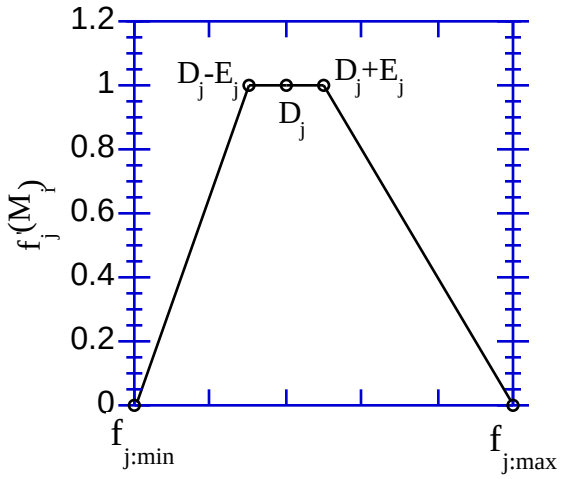

Figure 5. The fuzzy logic based fitness function.

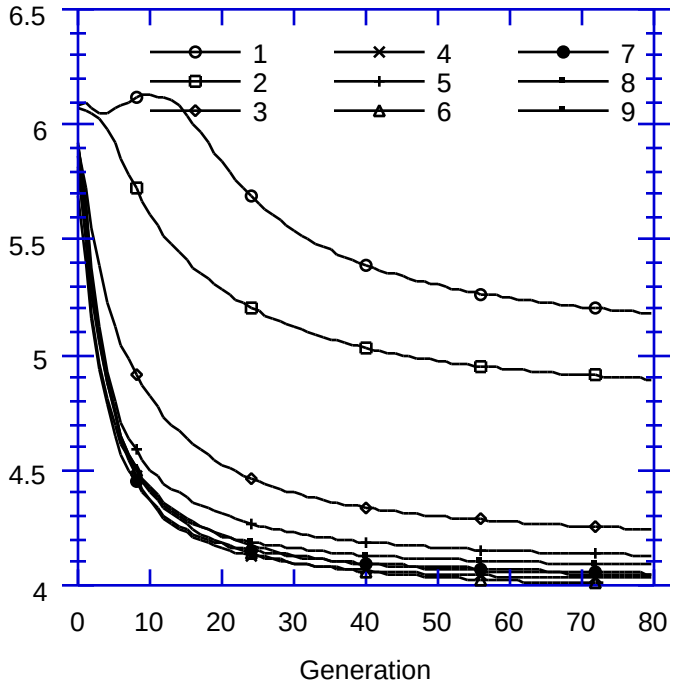

Figure 6. The <A> values as a function of generation and number of objectives.

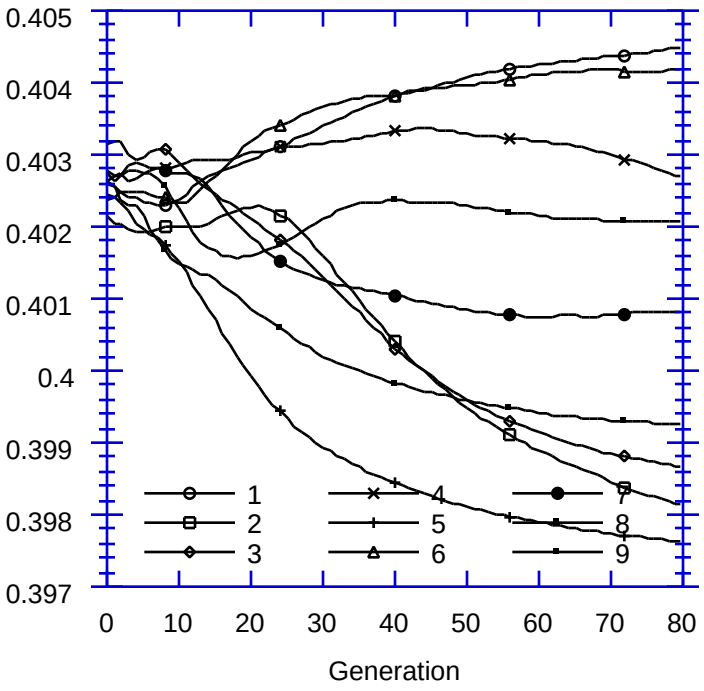

Figure 7. The <σ> values as a function of generation and number of objectives.

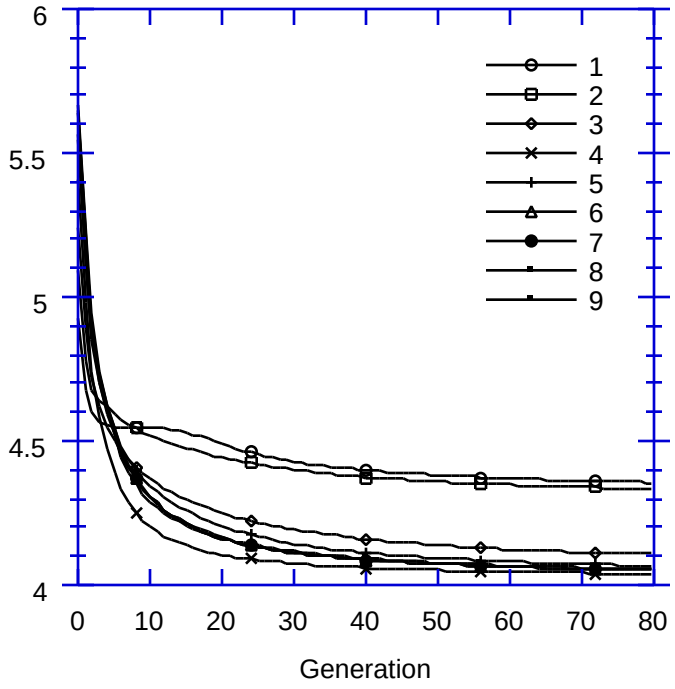

Figure 8. The <d> values as a function of generation and number of objectives.

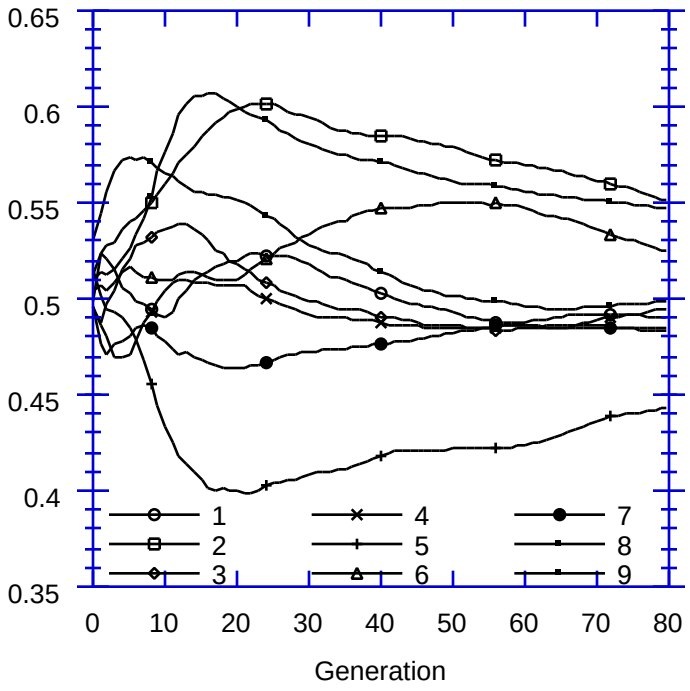

Figure 9. The <R> values as a function of generation and number of objectives.

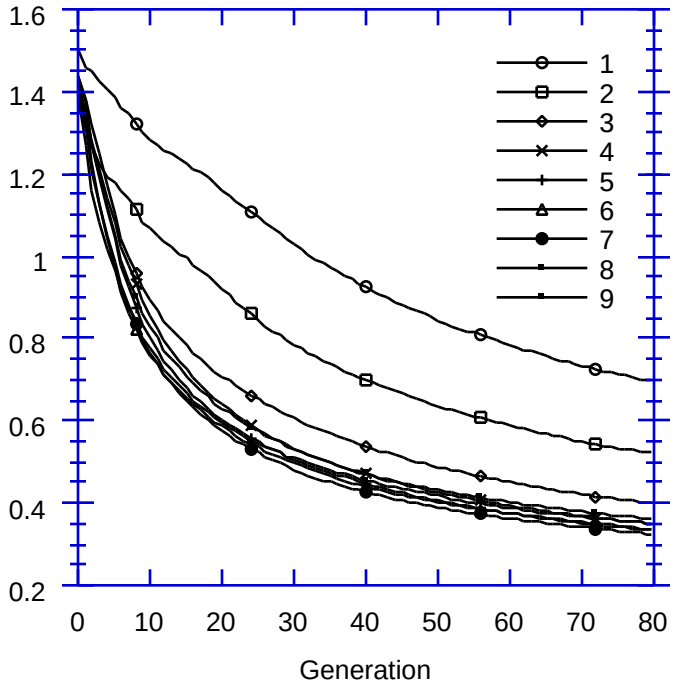

Figure 10. The standard deviations of A as a function of generation and number of objectives.

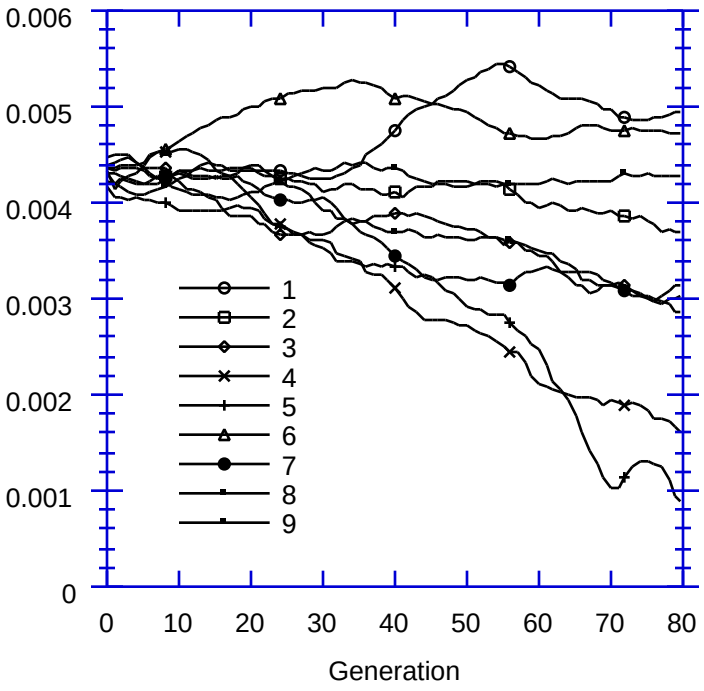

Figure 11. The standard deviations of σ as a function of generation and number of objectives.

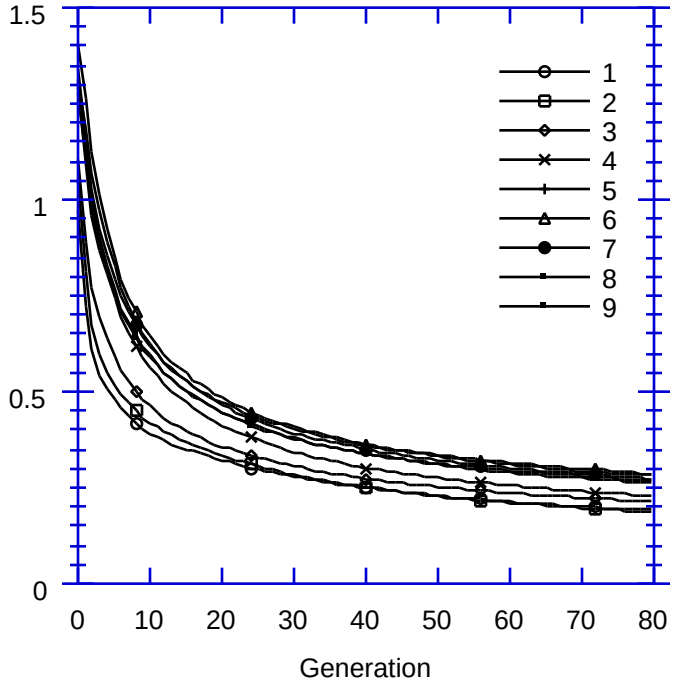

Figure 12. The standard deviations of d as a function of generation and number of objectives.

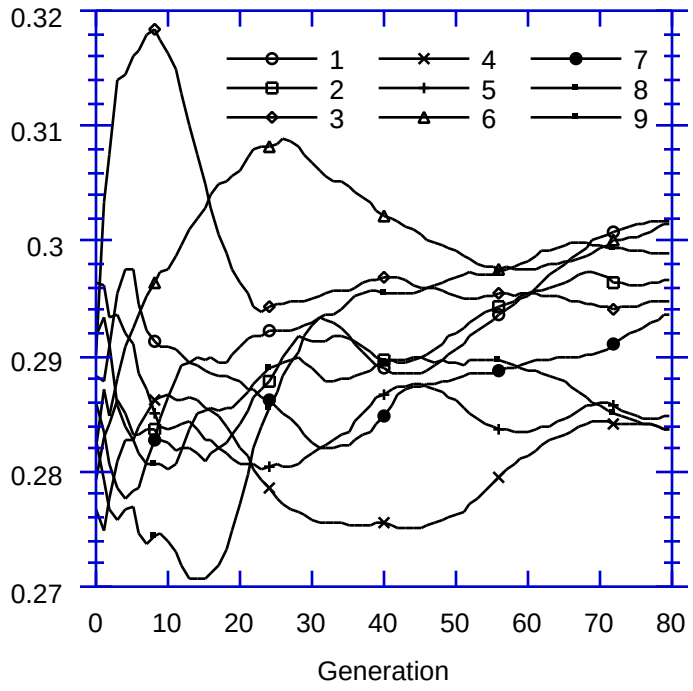

Figure 13. The standard deviations of R as a function of generation and number of objectives.

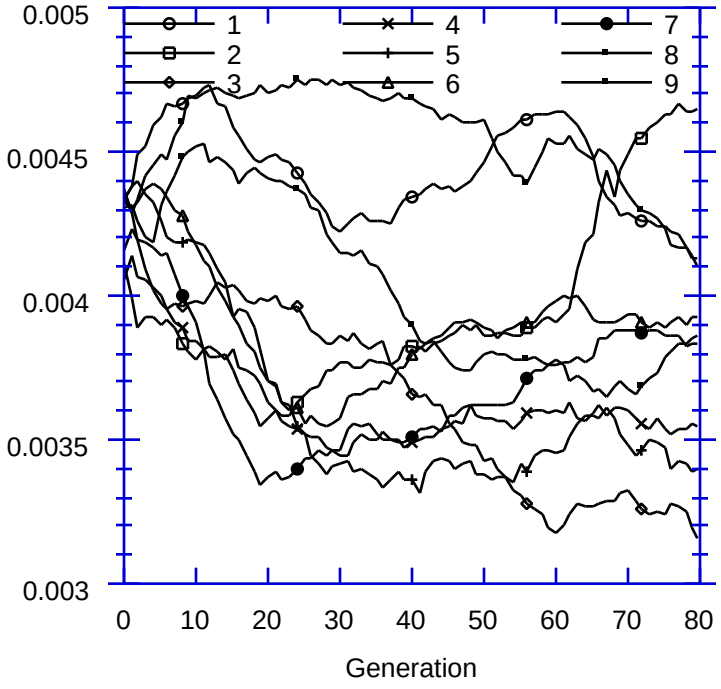

Figure 14. Eigenvalue #1, λ_1, as a function of generation and number of objectives.

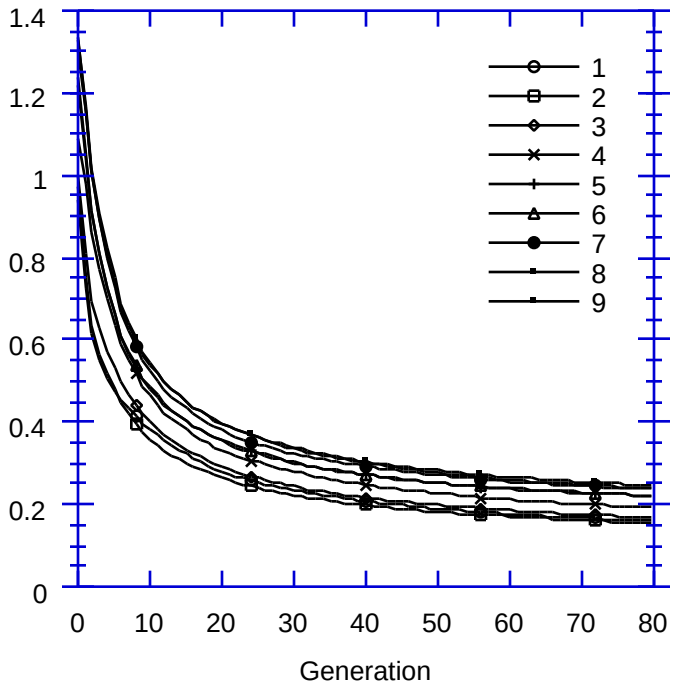

Figure 15. Eigenvalue #2, λ_2, as a function of generation and number of objectives.

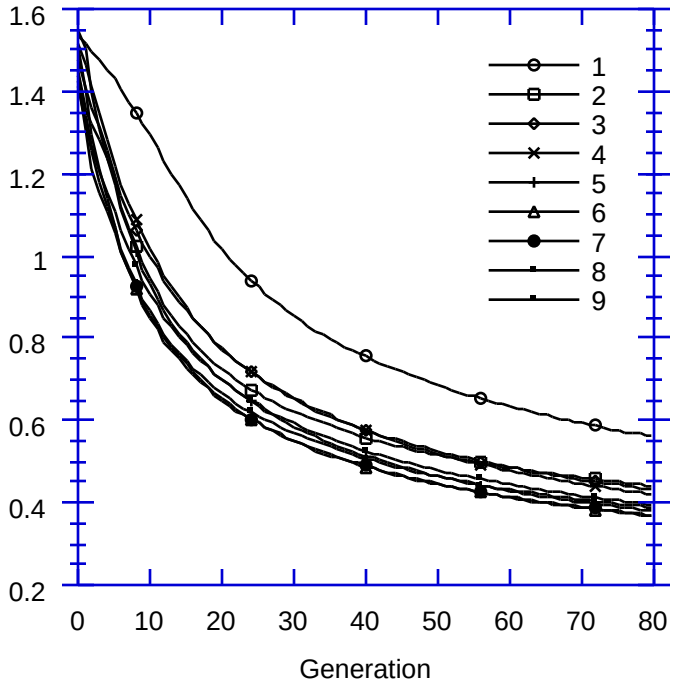

Figure 16. Eigenvalue #3, λ_3, as a function of generation and number of objectives.

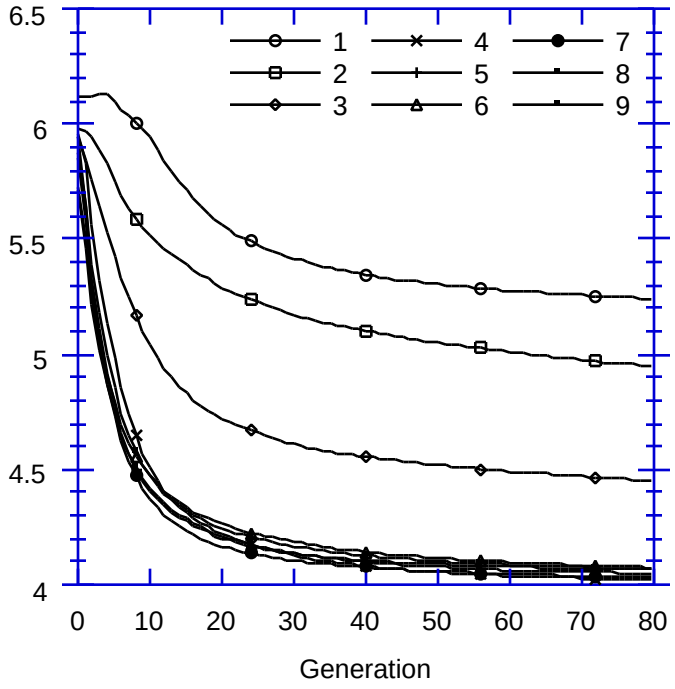

Figure 17. The <A> values as a function of generation and number of objectives.

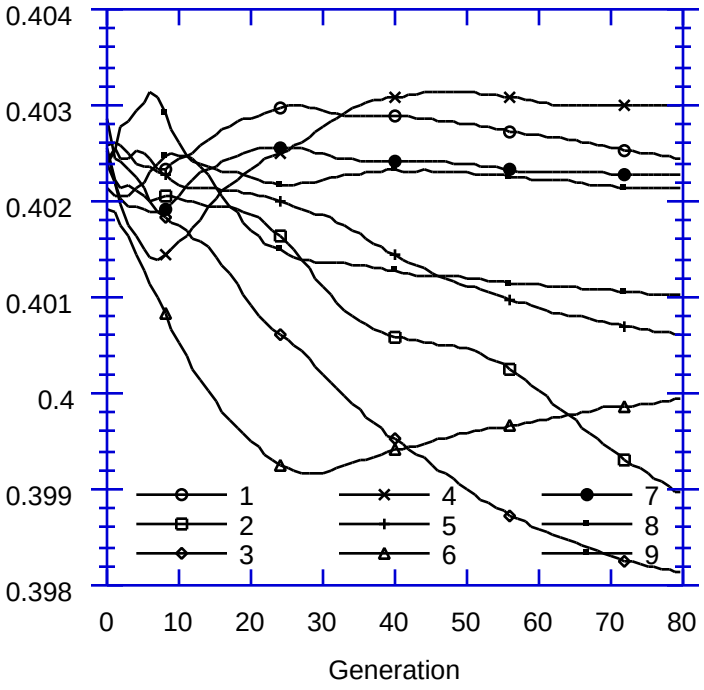

Figure 18. The <σ> values as a function of generation and number of objectives.

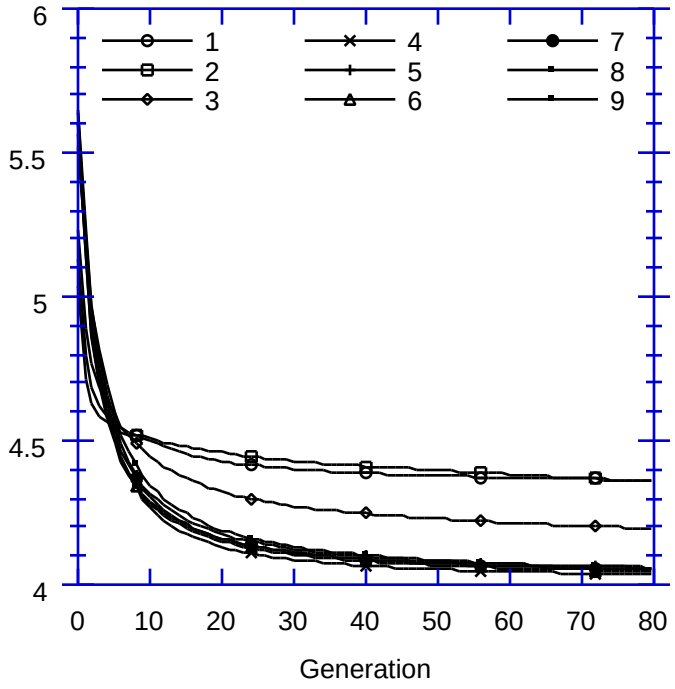

Figure 19. The <d> values as a function of generation and number of objectives.

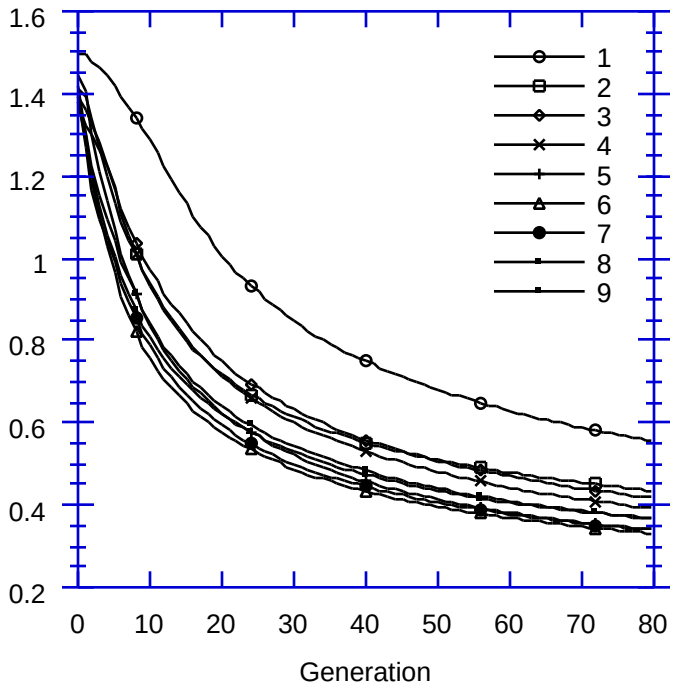

Figure 20. The standard deviations of A as a function of generation and number of objectives.

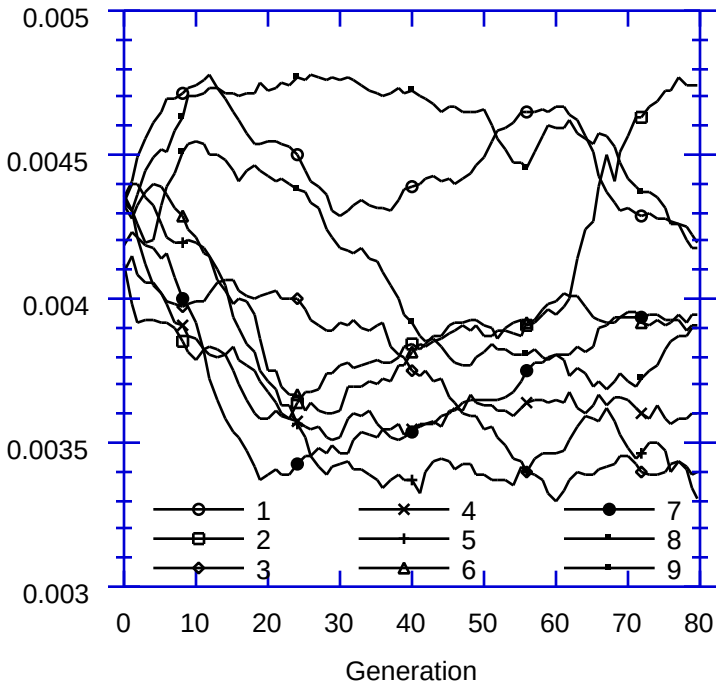

Figure 21. The standard deviations of σ as a function of generation and number of objectives.

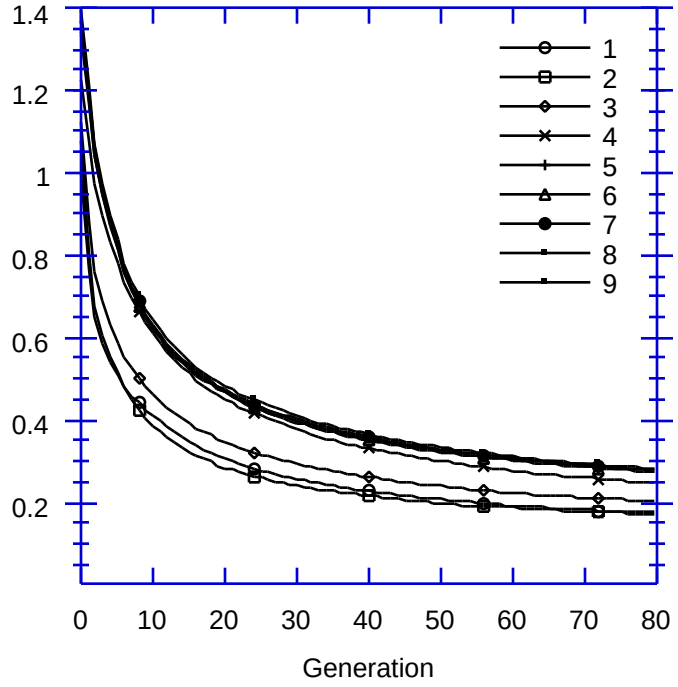

Figure 22. The standard deviations of d as a function of generation and number of objectives.